PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L — A2

(11) International Publication Number: **WO 99/44324**

(43) International Publication Date: 2 September 1999 (02.09.99)

(21) International Application Number: PCT/US99/04126

(22) International Filing Date: 25 February 1999 (25.02.99)

(30) Priority Data:
09/030,935 26 February 1998 (26.02.98) US

(71) Applicant: ARITHMETICA, INC. [US/US]; Corporation Trust Center, 1209 Orange Street, Wilimington, DE 19801 (US).

(72) Inventors: ANSHEL, Iris; 31 Peter Lynas Court, Tenafly, NJ 07670 (US). ANSHEL, Michael, M.; Apartment 3C, 1140 Fifth Avenue, New York, NY 10128 (US). GOLDFELD, Dorian; 31 Peter Lynas Court, Tenafly, NJ 07670 (US).

(74) Agents: KOCH, Robert, J. et al.; Fulbright & Jaworski L.L.P., 801 Pennsylvania Avenue, N.W., Washington, DC 20004 (US).

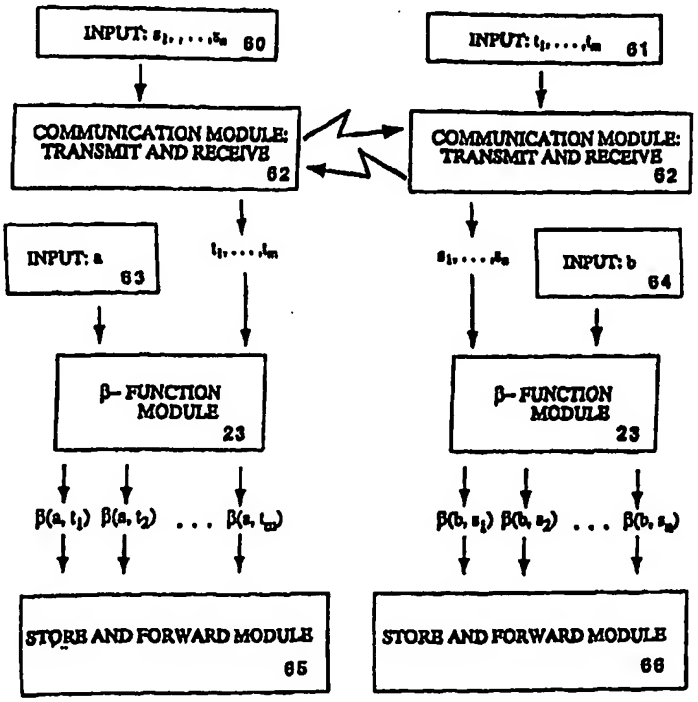
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: A METHOD AND APPARATUS FOR CRYPTOGRAPHICALLY SECURE ALGEBRAIC KEY ESTABLISHMENT PROTOCOLS

(57) Abstract

The present invention is a method and apparatus for providing cryptographically secure algebraic key establishment protocols that use monoids and groups possessing certain algorithmic properties. Special fast algorithms associated with certain monoids and groups are used to optimize both key agreement and key transport protocols. The cryptographic security of the algorithm is based on the difficulty of solving the conjugacy problem in groups and other known hard algebraic problems. Braid groups and their associated algorithms are the basis for highly rapid key agreement and key transport protocols which employ modest computational resources.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

A METHOD AND APPARATUS FOR CRYTOGRAPHICALLY SECURE ALGEBRAIC KEY ESTABLISHMENT PROTOCOLS

Inventors: Iris Anshel, Michael M. Anshel, Dorian Goldfeld

BACKROUND OF THE INVENTION

1. Field of the Invention

The invention relates to algebraic key establishment protocols for cryptographic applications.

2. Description of the Prior Art

Key Establishment Protocols

The concepts, terminology and framework for understanding cryptographic key establishment protocols is given in Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, "Handbook of Applied Cryptography," CRC Press (1997), pages 490–491.

A 'protocol' is a multi–party algorithm, defined by a sequence of steps specifying the actions required of two or more parties in order to achieve a specified objective.

A 'key establishment' protocol is a protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic applications.

A 'key transport' protocol is a key establishment protocol where one party creates or obtains a secret value, and securely transfers it to the other participating parties.

A 'key agreement' protocol is a key establishment protocol in which a shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of the participating parties such that no party can predetermine the resulting value.

A 'key–distribution' protocol is a key establishment protocol whereby the established keys are completely determined a priori by initial keying material.

A 'dynamic' key establishment protocol is one whereby the key established by a fixed

pair (or subset) of the participating parties varies on subsequent executions. Dynamic key establishment protocols are also referred to as 'session' key establishment protocols, and it is usually intended that these protocols are immune from known-key attacks.

The Diffie–Hellman key agreement protocol (also called 'exponential key exchange') is a fundamental algebraic protocol. It is presented in W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transaction on Information Theory vol. IT 22 (November 1976), pp. 644–654. The Diffie–Hellman key agreement protocol provided the first practical solution to the key distribution problem, allowing two parties, never having met in advance or sharing keying material, to establish a shared secret by exchanging messages over an open channel. The security rests on the intractability of the Diffie–Hellman problem and the related problem of computing discrete logarithms in the multiplicative group of the finite field GF(p) where p is a large prime, cf. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, "Handbook of Applied Cryptography," CRC Press (1997), page 113.

A key establishment protocol is said to have 'perfect forward secrecy' if compromise of long–term keys does not compromise past session keys. The idea of perfect forward security is that previous traffic is locked safely in the past. It may be provided by generating session keys by Diffie–Hellman key agreement, wherein the Diffie–Hellman exponentials are based on short term keys. If long–term secret keys are compromised, future sessions are nonetheless subject to impersonation by an active adversary (cf. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, "Handbook of Applied Cryptography," CRC Press (1997), page 496).

'Point–to–point key update' techniques based on symmetric encryption would make use of a long–term symmetric key K shared a priori by two parties A and B. The Diffie–Hellman key agreement protocol allows for the establishment of such a K. Thus, the Diffie–Hellman key agreement protocol together with the symmetric encryption system provide the primitives in specifying a key transport protocol (cf. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, "Handbook of Applied Cryptography," CRC Press (1997), page 497).

Combinatorial Group Theory

The definition of a monoid is given in Serge Lang, "Algebra," Third Edition, Addison-Wesley Publishing Company Inc. (1993), page 3.

QUOTE

Let S be a set. A mapping $S \times S \longrightarrow S$ is sometimes called a law of composition (of S into itself). If x, y are elements of S, the image of the pair (x, y) under the mapping is also called their product under the law of composition, and will be denoted xy. . .

Let S be a set with a law of composition. If x, y, z are elements of S, then we may form their product in two ways: $(xy)z$ and $x(yz)$. If $(xy)z = x(yz)$ for all x, y, z in S then we say that the law of composition is associative.

An element e of S such that $ex = x = xe$ for all $x \in S$ is called a unit element. A unit element is unique, for if e' is another unit element, we have $e = ee' = e'$ by assumption. In most cases, the unit element is written simply 1 (instead of e). . .

A monoid is a set G, with a law of composition which is associative, and having a unit element (so that in particular, G is not empty).

UNQUOTE

The definition of a group is given in Serge Lang, "Algebra," Third Edition, Addison-Wesley Publishing Company Inc. (1993), page 7.

QUOTE

A group G is a monoid, such that for every element $x \in$ G there exists an element $y \in$ G such that $xy = yx = e$. Such an element y is called an inverse for x. Such an inverse is unique. . . We denote this inverse by x^{-1}.

UNQUOTE

The basic reference for concepts, terminology, and historical framework in combinatorial group theory is the monograph by Bruce Chandler and Wilhelm Magnus, "The history of combinatorial group theory: a case study in the history of ideas," Springer-Verlag (1982). We quote from page 3:

QUOTE

Combinatorial group theory may be characterized as the theory of groups which are given by generators and defining relations, or, as we would say today, by a presentation.

UNQUOTE

The following problems were posed by M. Dehn in 1911. We quote from the monograph by Bruce Chandler and Wilhelm Magnus, "The history of combinatorial group theory: a case study in the history of ideas," Springer–Verlag (1982), page 19.

QUOTE

The Word Problem (called *Identitaetsproblem* by Dehn) Let an arbitrary element of the group be given through its buildup in terms of the generators. Find a method to decide in a finite number of steps whether this element equals the identity element or not.

The Conjugacy Problem (called *Transformationsproblem* by Dehn) Any two elements S and T of the group are given. Find a method to decide whether S and T are conjugate, i.e. whether there exists an element U of the group which satisfies the relation $S = UTU^{-1}$.

UNQUOTE

The comparison form of the word problem can be stated as follows:

Comparison Form of the Word Problem Let u, v be any two elements of the group given. Find a method to decide in a finite number of steps whether $u = v$.

Assume that G is a group given by a presentation $P(\mathrm{G})$. Let $W(\mathrm{G})$ denote the set of all words in the generators and their inverses given in the presentation of G. The functional form of the word problem is to produce a mapping F from $W(\mathrm{G})$ to $W(\mathrm{G})$ such that for all $u, v \in W(\mathrm{G})$ it follows that $F(u) = F(v)$ if and only if u, v define the same element of G with respect to the presentation $P(\mathrm{G})$. For each element $u \in W(\mathrm{G})$ the element $F(u)$ is termed the canonical form of u.

The functional form of the word problem requires an algorithm to produce canonical forms.

The Canonical Form Problem Let u be an arbitrary element of the given group. Specify a method to find, in a finite number of steps, a canonical form for u.

The functional form of the conjugacy problem requires, in addition, an algorithm to

actually produce the conjugating element U.

Generalized Conjugacy Problem (functional form) Let $s_1, s_2, \ldots, s_n$ be elements of a group G. Assume that $a \in G$ is secret and the set of n pairs of elements of the group G

$$\{s_1,\ a^{-1}s_1a\},\ \{s_2,\ a^{-1}s_2a\},\quad \ldots \quad \{s_n,\ a^{-1}s_na\}$$

are publicly announced. Find an algorithm to actually produce such an element a.

It is self evident that this problem is harder than the original conjugacy problem. It has been known for some time that there exist groups with solvable word problem and unsolvable conjugacy problem. For example, in D. J. Collins and C. F. Miller III, "The conjugacy problem and subgroups of finite index," Proc. LMS Series 3, 34, (1977), p. 535–556) it is shown that there exist finitely presented groups G with solvable word problem which contain a subgroup H of index 2 with an unsolvable conjugacy problem. (Of course, the word problem for H is solvable.)

The discrete logarithm problem for a finite cyclic group of order p (a large prime) provides a bridge from combinatorial group theory to cryptographic protocols. A finite cyclic group of order p can be realized as the set of integers coprime to p modulo p, i.e., the finite set of integers $\{1, 2, \ldots, p-1\}$ which forms a group under multiplication modulo p. Given fixed integers $a, b \in \{1, 2, \ldots, p-1\}$, where a is a primitive root modulo p, the discrete logarithm problem is to find an integer x (with $1 \leq x \leq p-1$) such that

$$b = a^x \pmod{p}.$$

Another realization of a finite cyclic group of order p can be specified by a presentation with one generator a and one defining relation $a^p = 1$ where 1 denotes the identity element. Note that every element g of the group has a unique canonical form $g = a^x$ where x is an integer between one and p. It is clear that the discrete logarithm problem for a finite cyclic group of order p is thus identical to the canonical word problem for this group with respect to an arbitrary primitive element a.

The present invention employs the problems and algorithms of combinatorial group to create novel cryptographically secure algebraic key establishment protocols. More specifically, the cryptographic security of these protocols depend on the existence of groups with

feasible word problem and hard conjugacy problem. Such an approach does not exist in the prior art.

SUMMARY OF THE INVENTION

It is the primary object of the present invention to provide novel cryptographically secure algebraic key establishment protocols based on a key establishment algebraic system KEAS.

Let $(\mathbf{U}, \theta_{\mathbf{U}})$ denote a monoid whose generating set $\{u_1, u_2, \dots\}$ is enumerable and whose law of composition

$$\theta_{\mathbf{U}} : \mathbf{U} \times \mathbf{U} \longrightarrow \mathbf{U}$$

is feasibly computable. Let $(\mathbf{V}, \theta_{\mathbf{V}})$ denote another such monoid. A KEAS is a five–tuple $(\mathbf{U}, \mathbf{V}, \beta, \gamma_1, \gamma_2)$ where

$$\beta : \mathbf{U} \times \mathbf{U} \longrightarrow \mathbf{V}, \gamma_i : \mathbf{U} \times \mathbf{V} \longrightarrow \mathbf{V} \quad (i = 1, 2)$$

are feasibly computable functions satisfying the following properties.

(i) For all elements $x, y_1, y_2 \in \mathbf{U}$

$$\beta(x, \theta_{\mathbf{U}}(y_1, y_2)) = \theta_{\mathbf{V}}(\beta(x, y_1), \beta(x, y_2))$$

(ii) For all elements $x, y \in \mathbf{U}$

$$\gamma_1(x, \beta(y, x)) = \gamma_2(y, \beta(x, y)).$$

It is an object of the present invention to provide an apparatus which can perform monoid multiplication for KEAS.

It is an object of the present invention to provide a novel algebraic key agreement protocol based on KEAS $= (\mathbf{U}, \mathbf{V}, \beta, \gamma_1, \gamma_2)$ where $\mathbf{U} = \mathbf{V} = \mathbf{G}$ is a group.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key agreement protocol whose security is based on the existence of groups whose word problem can be solved in polynomial time while no polynomial time algorithm to solve the generalized conjugacy problem is known.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key agreement protocol which is based on the computation of a list of randomly

rewritten conjugates in a group, thus reducing the steps and calculations in executing the protocol. This allows for easy implementation of the algorithms on low level computing devices with table driven modules.

It is an object of the present invention to provide an algebraic key agreement protocol based on KEAS = $(U, V, \beta, \gamma_1, \gamma_2)$ where $U = V = G$ is the braid group.

It is an object of the present invention to provide an apparatus which randomly rewrites a word in the braid group in linear time in the word length.

A key transport protocol is an algorithm, initiated by an input, defined by a sequence of steps, which enables one party to securely transfer a key to another party. The key transport protocol is said to run in polynomial time if the number of steps required to transfer the key is a polynomial in the bit length of the input. If the polynomial is of the first degree, the key transport protocol is said to run in linear time.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key transport protocol based on KEAS which allows for a linear time secure transfer of an encrypted key and requires polynomial time decryption of said encrypted key.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key transport protocol based on KEAS = $(U, V, \beta, \gamma_1, \gamma_2)$ where U and V are monoids and U acts on a message space. The key transport protocol is a combination of the algebraic key agreement protocol based on KEAS, together with an apparatus which efficiently compares members of the message space. This allows for linear time secure transfer of an element of the message space and requires a polynomial time algorithm for comparison and retrieval of the message.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key transport protocol based on KEAS = $(U, V, \beta, \gamma_1, \gamma_2)$ where the message space = $U = V$ is the braid group which acts on itself by multiplication. This allows for the linear time secure transfer of an element of the message space (randomly rewritten word in the braid group) and requires a polynomial time algorithm to obtain a canonical form and decrypt the message.

It is an object and feature of the present invention to provide a cryptographically secure algebraic key transport protocol based on KEAS = $(U, V, \beta, \gamma_1, \gamma_2)$ where the message space = U = V is the braid group which acts on itself by conjugation. This allows for the linear time secure transfer of an element of the message space (randomly rewritten word in the braid group) and requires a polynomial time algorithm to obtain a canonical form and decrypt the message.

It is an object of the present invention to provide a cryptographically secure algebraic key transport protocol based on KEAS = $(U, V, \beta, \gamma_1, \gamma_2)$ where U = V is the braid group and the message space is a free group.

The system according to the invention is particularly suited towards implementation using currently available digital technology, commercially popular microprocessor based systems, and other affordable digital components. Significant portions of the system may be implemented and significant portions of the method according to the invention may be performed by software in a microcomputer based system. Moreover the system is quite suitable for implementation on emerging computer technologies, e.g., quantum computers.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows an exemplary preferred embodiment of an apparatus which performs monoid multiplication.

Fig. 2 shows a β–function module.

Fig. 3 shows a γ_1, γ_2–function module.

Fig. 4 shows a submonoid generator

Fig. 5 shows a submonoid random element generator.

Fig. 6 shows an exchange of public information of an algebraic key agreement protocol based on monoids.

Fig. 7 shows a preferred embodiment of an apparatus which performs the algebraic key agreement protocol based on monoids.

Fig. 8 shows an exchange of public information of an algebraic key agreement protocol based on the braid group.

Fig. 9 shows a preferred embodiment of a random rewriter for the braid group

Fig. 10 shows a preferred embodiment of an apparatus which performs the algebraic key agreement protocol based on the braid group.

Fig. 11 shows a preferred embodiment of an apparatus which performs the algebraic key transport protocol for monoids.

Fig. 12 shows a preferred embodiment of an apparatus which performs the algebraic key transport protocol for groups.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A General Algebraic Key Agreement Protocol

A preferred embodiment of an apparatus which performs an algebraic key agreement protocol based on KEAS will now be described in detail. The algebraic key-agreement protocol requires an apparatus which can perform monoid multiplication. A preferred exemplary embodiment of such an apparatus is depicted with block diagrams in figure 1, and is described as follows.

Let $(\mathrm{U}, \theta_{\mathrm{U}})$ denote a monoid whose generating set $\{u_1, u_2, \dots\}$ is enumerable and whose law of composition

$$\theta_{\mathrm{U}} : \mathrm{U} \times \mathrm{U} \longrightarrow \mathrm{U}$$

is feasibly computable. The U-Library 11 consists of the set of generators $\{u_1, u_2, \dots\}$. A sequence of indices 10 along with the U-Library 11 is presented to the Sequence Encoder 12. The Sequence Encoder chooses 12 $u_{i_1}, u_{i_2}, \dots, u_{i_e}$ from the U-library 11 and presents this to the Free Monoid Multiplier 13 which then concatenates the elements to yield the output $u_{i_1} \cdot u_{i_2} \cdots u_{i_e}$. The monoid U can be viewed as a quotient of the free monoid (generated by the U-Library) modulo a set of rewriting rules. The U-Presentation Code 14 consists of this set of rewriting rules. The Monoid Rewriter 15 computes the equivalence

class of $u_{i_1} \cdot u_{i_2} \cdots u_{i_a}$ modulo the rewriting rules in the U–Presentation Code 14. The result is a word in the monoid U. An apparatus which performs the internal binary operation of U can now be specified. Given $x = u_{j_1} \cdot u_{j_2} \cdots u_{j_a}$, and $y = u_{k_1} \cdot u_{k_2} \cdots u_{k_b}$, to obtain the product $x \cdot y$, simply input the long sequence $j_1, j_2, \ldots j_a, k_1, k_2, \ldots k_b$ into 10. The output of the Monoid Rewriter 15 will be $x \cdot y$.

A preferred embodiment of an apparatus which performs the algebraic key–agreement protocol based on KEAS is depicted in block diagrams in figures 1 through 7. Recall that a KEAS is a five–tuple $(\mathbf{U}, \mathbf{V}, \beta, \gamma_1, \gamma_2)$ where U and V are monoids with feasibly computable laws of composition and $\beta, \gamma_1, \gamma_2$ are functions satisfying the following properties:

(i) For all $x, y_1, y_2 \in \mathbf{U}$

$$\beta(x, \theta_{\mathbf{U}}(y_1, y_2)) = \theta_{\mathbf{V}}(\beta(x, y_1), \beta(x, y_2))$$

(ii) There exists easily computable functions $\gamma_i : \mathbf{U} \times \mathbf{V} \longrightarrow \mathbf{V}$ $(i = 1, 2)$ such that

$$\gamma_1(x, \beta(y, x)) = \gamma_2(y, \beta(x, y)).$$

Let $x, y \in \mathbf{U}$ denote the Input 20. The β–Function Module 21 computes the value of $\beta(x, y)$. Let $u \in \mathbf{U}$, be the Input 30, and let $v \in \mathbf{V}$ be the Input 31. The γ_1–Function Module 32 computes $\gamma_1(u, v)$ while the γ_2–Function Module 32 computes $\gamma_2(u, v)$.

With the functions $\beta, \gamma_1, \gamma_2$ in place the algebraic key agreement protocol can now be described. Given a subset $\mathbf{S} \subseteq \mathbf{U}$, recall that the submonoid generated by S, denoted $< \mathbf{S} >$, is defined to be the smallest submonoid of U which contains S, i.e.,

$$< \mathbf{S} > = \cap \text{ submonoids of } \mathbf{U} \text{ containing } S.$$

Remark that $< \mathbf{S} >$ coincides with the set of all possible products in products of elements in the set S, including the empty product (which is the identity element).

The algebraic key agreement protocol involves two users, A(lice) and B(ob). User A is assigned a set of distinct elements in U,

$$\{s_1, \ldots, s_n\}$$

which generate a submonoid of U denoted

$$\mathbf{S_A} = < s_1, s_2, \ldots, s_n > .$$

In the discussion below we utilize functional notation for elements in the monoid U: if x is an element in U, x is expressible as a word in the generators of U and we write

$$x = x(u_{i_1}, u_{i_2}, \ldots, u_{i_n}).$$

Remark that each s_i is expressible as a word in the generators of U: for $i = 1, 2, \ldots, n$,

$$s_i = s_i(u_{i_1}, u_{i_2}, \ldots).$$

Likewise user B is assigned elements $\{t_1, \ldots, t_m\}$ which generate a submonoid of U denoted

$$\mathrm{T_B} = < t_1, t_2, \ldots, t_m > .$$

Here again each t_j is expressible as a word in the generators of U: for $j = 1, 2, \ldots, m$,

$$t_j = t_j(u'_{j_1}, u'_{j_2}, \ldots).$$

An apparatus for assigning an arbitrary set $w_1, w_2, \ldots w_m$ of m words to a user is depicted in figure 4. The key component of this apparatus is a cryptographically secure pseudorandom number generator PRNG. The definition of a PRNG is given in Bruce Schneier, "Applied cryptography protocols, algorithms, and source code in C," Second Edition 1996, John Wiley, page 45, and is well known in the art. In all subsequent discussions in the preferred embodiment, a PRNG will always refer to such a cryptographically secure pseudorandom number generator.

Let $m, k \geq 1$ denote integers. Let $L = \{L_1, L_2, \ldots L_m\}$ denote a vector of positive integers. The Input: m, L 40 together with the Input: k 42 is presented to a pseudorandom number generator PRNG 41 which creates m lists of integers of lengths $L_1, L_2, \ldots L_m$, respectively; each list $\{e(i,1), e(i,2), \ldots, e(i,L_i)\}$ (for $i = 1, 2, \ldots m$) consisting of integers randomly chosen from the set $\{1, 2, \ldots, k\}$. These lists, together with the U-Library 11 are then presented to the Sequence Encoder 12 whose output goes to the Free Monoid Multiplier 13. The output of the Free Monoid Multiplier 13 is then sent to the Monoid Rewriter 15 into which the U-presentation code has already been presented. The final output is $w_1, w_2, \ldots, w_m$ which creates a User Submonoid Generator Library 43 and then sent to the User Submonoid Store and Forward Module 44.

The process of key exchange begins with both users choosing secret elements in their

respective submonoids,

$$a \in S_A, \ a = a(s_1, s_2, \ldots, s_n)$$

$$b \in T_B, \ b = b(t_1, t_2, \ldots, t_m).$$

This is depicted in figure 5. Let L, m denote positive integers. The Input: L 50 together with the Input: m 52 is sent to a pseudorandom number generator PRNG 51 which randomly chooses $L' \leq L$ positive integers $e_1, e_2, \ldots, e_{L'}$ such that each $e_i \leq m$ (for $i = 1, 2, \ldots L'$). This sequence of randomly chosen integers is presented to the Sequence Encoder 12 which also receives the Input of the User Submonoid Generator Library 43 which consists of $w_1, w_2, w_3, \ldots$ The Sequence Encoder 12 then chooses $w_{e_1}, w_{e_2}, \ldots w_{e_{L'}}$ and presents this to the Submonoid Multiplier 54 which computes the product $a = w_{e_1} \cdot w_{e_2} \cdots w_{e_{L'}}$ and sends it to the User Private Element Store and Forward Module 55.

User A now transmits the Input 60

$$s_1, s_2, \ldots, s_n$$

(where each s_i is a word in the generators of U) via the Communication module: Transmit and Receive 62, and user B transmits the Input 61

$$t_1, t_2, \ldots, t_m$$

via the Communication module: Transmit and Receive 62. The received list $\{t_1, t_2, \ldots, t_m\}$ together with Alice's secret key, the Input: a 63 is then forwarded to the β–Function Module 23 yielding the list

$$\beta(a, t_1), \ldots, \beta(a, t_m)$$

which is stored in the Store and Forward Module 65. Similarly, the received list $\{s_1, s_2, \ldots, s_n\}$ together with Bob's secret key, the Input: b 64 is then forwarded to the β–Function Module 23 yielding the list

$$\beta(b, s_1), \ldots, \beta(b, s_n)$$

which is stored in the Store and Forward Module 66.

User A now transmits the Input 70

$$\beta(a, t_1), \ldots, \beta(a, t_m)$$

(which was stored is the Store and Forward Module 65) via the Communication module: Transmit and Receive 62, and similarly user B transmits the Input 71

$$\beta(b, s_1), \ldots, \beta(b, s_n)$$

(which was stored is the Store and Forward Module 66) via the Communication module: Transmit and Receive 62.

The received list $\beta(b, s_1), \ldots, \beta(b, s_n)$, together with the secret list of integers $e_1, e_2, \ldots, e_{L'}$ generated by the PRNG 51 to produce Alice's secret key

$$a = s_{e_1} \cdot s_{e_2} \cdots s_{e_{L'}}$$

is presented to the V–Monoid Multiplier 72 which then (using property (i) that β satisfies) computes the product

$$\beta(b, a) = \beta(b, s_{e_1}) \cdot \beta(b, s_{e_2}) \cdots \beta(b, s_{e_{L'}}).$$

The element $\beta(b, a)$ together with the secret key a are sent to the γ_1–Function 32 to produce the final output

$$\gamma_1(a, \beta(b, a)).$$

In a completely analogous manner, the received list $\beta(a, t_1), \ldots, \beta(a, t_m)$, together with the secret list of integers $f_1, f_2, \ldots, f_{L''}$ generated by the PRNG 51 to produce Bob's secret key

$$b = t_{f_1} \cdot t_{f_2} \cdots t_{f_{L''}}$$

is presented to the V–Monoid Multiplier 72 which then (using property (i) that β satisfies) computes the product

$$\beta(a, b) = \beta(a, t_{f_1}) \cdot \beta(a, t_{f_2}) \cdots \beta(a, t_{f_{L''}}).$$

The element $\beta(a, b)$ together with the secret key b are sent to the γ_2–Function 33 to produce the final output

$$\gamma_2(b, \beta(a, b)).$$

By property (ii) it immediately follows that

$$\gamma_1(a, \beta(b, a)) = \gamma_2(b, \beta(a, b))$$

which is the common key exchanged between Alice and Bob.

Example 1:

A first example of an algebraic key agreement protocol of the type detailed above can be obtained by considering the case where $U = V = G$ is a group (a monoid where very element has an inverse). In this case the function β,

$$\beta : G \times G \longrightarrow G$$

is chosen to be conjugation:

$$\beta(x, y) = x^{-1} y\, x.$$

The functions γ_1 and γ_2 are chosen to be

$$\gamma_1(u, v) = u^{-1} v \qquad \gamma_2(u, v) = v^{-1} u.$$

It is easy to see that properties (i), (ii) hold.

The asymmetric key agreement protocol in this situation can be described as follows. Users A and B publicly choose subgroups

$$S_A = < s_1, s_2, \ldots, s_m > \qquad S_B = < t_1, \ldots, t_n >,$$

and secret elements $a \in S_A$ and $b \in S_B$. User A transmits the collection of conjugates

$$a^{-1} t_1\, a,\ a^{-1} t_2\, a,\ \ldots,\ a^{-1} t_m\, a$$

and similarly user B transmits

$$b^{-1} s_1\, b,\ b^{-1} t_2\, b,\ \ldots,\ b^{-1} s_n\, b$$

Recalling that the conjugate of the product of two elements is the product of the conjugates of those elements, users A and B are now in a position to compute, respectively, the elements

$$b^{-1} a\, b, \qquad a^{-1} b\, a.$$

In order to attain a common key, user A now multiplies $b^{-1}ab$ on the left by a^{-1} to obtain

$$[a, b] = a^{-1} b^{-1} a\, b,$$

and user B multiplies $a^{-1}b\ a$ on the left by b^{-1} to obtain $[b, a]$ and then computes the inverse of $[b, a]$ which is $[a, b]$. Note that this is consistent with the general system notation in that

$$[a, b] = \gamma_1(a, \beta(b, a)) = \gamma_2(b, \beta(a, b)).$$

The fact that there exist groups with solvable word problem and unsolvable conjugacy problem, shows that at least in principle, the key agreement protocol for groups may be algorithmically unbreakable. In practice, however, one really works with a computer with only a finite amount of memory and this is equivalent to working only with words of bounded length in the group G. Thus everything is reduced to a finite amount of computation, so from this point of view all these problems are decidable.

The above protocol is secure and feasible provided the group G has a feasibly solvable word problem and hard conjugacy problem. There are many groups, however, where the word problem can be solved in polynomial time (in the word length), while at the same time, there is no known polynomial time algorithm for solving the generalized conjugacy problem (functional form). An illustrative example of such a group is the braid group of N symbols.

The braid group was first systematically studied by Emil Artin, "Theorie der Zöpfe," Hamb. Abh. 4 (1925), pages 47–72. In that paper, the so called Artin generators $x_1, x_2, \ldots x_N$ for the Braid group of N symbols are introduced. They satisfy the relations

$$x_i x_j = x_j x_i, \qquad \text{if } |j-i| \geq 2 \text{ and } 1 \leq i, j \leq N$$

$$x_i x_{i+1} x_i = x_{i+1} x_i x_{i+1}, \qquad \text{if } 1 \leq i \leq N-1.$$

A preferred embodiment of an apparatus which performs the key agreement protocol for the braid group is depicted in block diagrams in figures 8 to 10. This apparatus will now be described in detail.

Users A and B wish to exchange keys via public discussion over an insecure channel. Fix G to be the braid group on N generators. User A randomly chooses elements $s_1, s_2, \ldots, s_n \in G$ (Input 80) and transmits them to user B via the Communication Module 62. Similarly, user B randomly chooses elements $t_1, t_2, \ldots, t_m$ (Input 81) and transmits them to user A via the Communication Module 62. It can be assumed that $s_1, s_2, \ldots, s_n$, $t_1, t_2, \ldots t_n$ are publicly known.

The Input: $s_1, s_2, \ldots, s_n$ 80 is sent to the Random Word Generator 82 which produces a word a which is a secret word in the generators $s_1, s_2, \ldots, s_n$. The process for doing

this is depicted in a more general setting in figure 5. The Input: $t_1, t_2, \ldots, t_m$ 81 is sent to the Random Word Generator 83 which produces a word b which is a secret word in the generators $t_1, t_2, \ldots, t_m$. The secret word a together with the generators $t_1, t_2, \ldots, t_m$ are then presented to the Braid Group Conjugation Module 84 which computes the list of conjugate elements

$$a^{-1}t_1a,\ a^{-1}t_2a,\ \ \ldots\ \ a^{-1}t_ma.$$

Similarly, the secret word b together with the generators $s_1, s_2, \ldots, s_n$ are then presented to the Braid Group Conjugation Module 84 which computes the list of conjugate elements

$$b^{-1}s_1b,\ b^{-1}s_2b,\ \ \ldots\ \ b^{-1}s_nb.$$

In both cases, these lists are then sent to the Random Rewriter 85 which randomly rewrites each word in the list. The randomly rewritten lists are then sent to the Store and Forward Modules 86, 87.

A preferred embodiment of the Random Rewriter 85 is depicted in block diagrams in figure 9. The Input: w 90 is sent to the Free Reducer 91. The Free Reducer 91 searches for subwords of the form $x\,x^{-1}$ and $x^{-1}x$ in the word w (where x is an arbitrary word in the Artin generators of G) and replaces xx^{-1} and $x^{-1}x$ by the identity element. The Free Reducer 91 freely reduces the word w to produce the (possibly shorter) word W. The word W is then presented to the Length Function which computes its length L. The length L is then sent to a pseudorandom number generator PRNG 94 which randomly produces an integer j (where $1 \leq j \leq L$) and a bit e which is either 0 or 1. The freely reduced word W together with the integer j and the bit e are then sent to the Move and Replace Module 92 which produces a new word W' in the following manner.

Recall that W is a word in the Artin generators $x_1, x_2, \ldots, x_N$ of length L, say $W = x_{r_1}^{e_1} \cdot x_{r_2}^{e_2} \cdots x_{r_L}^{e_L}$ where for $i = 1, 2, \ldots, L$ each $e_i = \pm 1$ and $r_i \in \{1, 2, \ldots, N\}$. If $e = 0$ and $j = 1$, halt the process. If $e = 0$ and $j > 1$ consider the subword (of length 2 at the j^{th} position) $x_{r_{j-1}}^{e_{j-1}}\, x_{r_j}^{e_j}$. If $|r_{j-1} - r_j| \geq 2$ replace this subword by $x_{r_j}^{e_{j-1}}\, x_{r_{j-1}}^{e_{j-1}}$ and set j = j-1. Keep repeating until either $j = 1$ or $|r_{j-1} - r_j| = 1$. If $|r_{j-1} - r_j| = 1$, replace the string $x_{r_{j-1}}^{e_{j-1}}\, x_{r_j}^{e_j}$ by a four symbol subword arising from the Artin relations. The complete list of

substitutions is given as:

$$x_j\, x_{j+1} \longrightarrow x_{j+1}\, x_j\, x_{j+1}\, x_j^{-1}$$

$$x_j\, x_{j+1}^{-1} \longrightarrow x_{j+1}^{-1}\, x_j^{-1}\, x_{j+1}\, x_j$$

$$x_j^{-1}\, x_{j+1} \longrightarrow x_{j+1}\, x_j\, x_{j+1}^{-1}\, x_j^{-1}$$

$$x_j^{-1}\, x_{j+1}^{-1} \longrightarrow x_{j+1}\, x_j^{-1}\, x_{j+1}^{-1}\, x_j^{-1}$$

$$x_{j+1}\, x_j \longrightarrow x_j^{-1}\, x_{j+1}\, x_j\, x_{j+1}$$

$$x_{j+1}\, x_j^{-1} \longrightarrow x_j^{-1}\, x_{j+1}^{-1}\, x_j\, x_{j+1}$$

$$x_{j+1}^{-1}\, x_j \longrightarrow x_j\, x_{j+1}\, x_j^{-1}\, x_{j+1}^{-1}$$

$$x_{j+1}^{-1}\, x_j^{-1} \longrightarrow x_j\, x_{j+1}^{-1}\, x_j^{-1}\, x_{j+1}^{-1}$$

In an analogous manner if $e = 1$ the algorithm is the same except that one now considers the subword $x_{r_j}^{e_j}\, x_{r_{j+1}}^{e_{j+1}}$ and set $j = j + 1$. So if $e = 0$, move to the left; while if $e = 1$, move to the right searching for two adjacent generators whose indices differ by one. As soon as they are found, they are replaced according to the substitutions listed above.

The output W' of the Move and Replace Module 92 together with the Input 95 of a positive integer k is then sent to the Iterate and Exit Module 96 which iterates the above procedure k times (by sending W' back to the Free Reducer 91) and then exits the procedure sending its output W' to the Free Reducer 91. The final freely reduced word is then sent to the Store and Forward Module 97.

The list $a^{-1}t_1a,\ a^{-1}t_2a,\ \ldots\ a^{-1}t_ma$, which was stored in the Store and Forward Module 86 becomes Input 100 and is presented to the Communication Module: Transmit and Receive 62. Likewise the list $b^{-1}s_1b,\ b^{-1}s_2b,\ \ldots\ b^{-1}s_nb$, which was stored in the Store and Forward Module 87 becomes Input 101 and is presented to the Communication Module: Transmit and Receive 62. These lists are broadcast over an insecure channel and can be assumed to be publicly known. The received list $b^{-1}s_1b,\ b^{-1}s_2b,\ \ldots\ b^{-1}s_nb$ together with user A's secret word a are then presented to the Braid Group Multiplier 102 which computes $b^{-1}a\,b$. The conjugate $b^{-1}a\,b$ together with user A's secret word a is sent to the γ_1–Function 103 which computes the final output $a^{-1}b^{-1}a\,b$. Correspondingly,

the received list $a^{-1}t_1a$, $a^{-1}t_2a$, ... $a^{-1}t_ma$ together with user B's secret word b are then presented to the Braid Group Multiplier 102 which computes $a^{-1}b\,a$. The conjugate $a^{-1}b\,a$ together with user B's secret word b is sent to the γ_2–Function 104 which computes the final output $a^{-1}b^{-1}a\,b$ which is the exchanged key.

The total running time of this protocol will be polynomial time in the total bit length of the exchanged lists:

$$\{b^{-1}s_1b, b^{-1}s_2b, \ldots, b^{-1}s_nb\}, \quad \{a^{-1}t_1a, a^{-1}t_2a, \ldots, a^{-1}t_ma\},$$

provided the integer k (Input 95, which counts the number of iterations used by the random rewriter) is not too large.

A General Algebraic Key Transport Protocol

A preferred embodiment of an apparatus which performs the general algebraic key transport protocol will now be described in detail. It is assumed that two parties A(lice) and B(ob) have already participated in an algebraic key agreement protocol of the type described previously (for monoids), so that both A and B are in possession of a common key k which is a word in the monoid U. Note that the common key k may be expressed as a word in the generators of U in many different ways. Each such expression is contained in the same equivalence class of the free monoid modulo the presentation code of U. In order to obtain a unique expression for k it is necessary to have a unique canonical form for all elements in U. In the key transport protocol which will now be presented, it is not assumed that k is in canonical form.

The key transport protocol for monoids is based on the action of the monoid on a set M which we term the message space. The action of the monoid U on M is a function

$$\mathrm{U} \times \mathrm{M} \longrightarrow \mathrm{M}$$

which we denote

$$(u, m) \mapsto u(m) \in \mathrm{M}$$

for each $u \in \mathrm{U}, m \in \mathrm{M}$, which satisfies the following conditions:

$$u(v(m)) = uv(m), \qquad \text{for all } u, v \in \mathrm{U} \text{ and } m \in \mathrm{M}$$

$$1(m) = m, \qquad \text{for all } m \in \mathrm{M}.$$

A preferred embodiment of an apparatus which performs the key transport protocol for monoids is depicted in block diagrams in figure 11. First, a common key

$$k = \gamma_1(a, \beta(b, a)) = \gamma_2(b, \beta(a, b)) \in \mathrm{U}$$

is exchanged via the Key Agreement Protocol for Monoids 110 which was previously depicted in figures 6 and 7. Fix distinct elements

$$M_1, M_2, \ldots, M_D \in \mathrm{M}$$

which is Input 112. The key transport protocol is a mechanism which allows A (the sender) to transfer a message

$$M \in \{M_1, M_2, \ldots, M_D\}$$

to B (the receiver). The message M is the Input 111 which is sent to the Monoid Action 113 which computes the action of k on M which is $k(M)$. The element $k(M)$ is then sent to the Communication Module: Transmit and Receive 62 which transmits $k(M)$ to B. Concurrently, the Input: $M_1, M_2, \ldots, M_D \in \mathrm{M}$ 112 together with the key k (output of the Key Agreement Protocol for Monoids 110) is presented to the Monoid Action 113 which computes the elements $k(M_1), k(M_2), \ldots, k(M_D)$. These elements together with $k(M)$ is presented to the Compare and Choose Module 114 which compares them and determines which of the M_i for $i = 1, 2, \ldots, D$ is actually M. Thus, the message M has been transferred from A(lice) to B(ob).

Note that in this protocol, it is not necessary to compute canonical forms. All that is required is an algorithm to decide (Compare and Choose Module 114) if two elements of the set M are the same or not.

In a key transport protocol, the bandwidth is defined to be the number of bits publicly exchanged between the two parties (via the Communication Module: Transmit and Receive 62) in order to transmit one bit (shared secret). In this protocol, the bandwidth decreases

as D increases, but at the expense of an increase in off line computations (Compare and Choose Module 114).

In a binary system such as for digital message transmissions, the input 112 may be arbitrarily selected to be one of two elements M_1 or M_2. The monoid action may be conducted on the single arbitrary selected element. If the output of the monoid action 113 matches the input $k(M)$, then $k(M)$ may be presumed to represent the selected element. If $k(M)$ does not match the output of the monoid action 113, then M is presumes to be the non–selected element.

If our module U is a group G then the basic property of a group (that every element has a unique inverse) allows us to present a different key transport protocol. It is again assumed that two parties A(lice) and B(ob) have already participated in an algebraic key agreement protocol of the type described previously (for groups), so that both A and B are in possession of a common key k which is a word in the group G. It is not assumed that k is in canonical form.

A preferred embodiment of an apparatus which performs the key transport protocol for groups is depicted in block diagrams in figure 12. Let $k \in G$, be the common key exchanged via the Key Agreement Protocol for Groups 120 which was previously depicted in figures 8 and 9. Let $M \in$ M be the Input 121. This is sent to the Group Action 123 which computes $k(M)$ which is transmitted to B(ob) via the Communication Module: Transmit and Receive 62. Concurrently, the common key k which is the output of the Key Agreement Protocol for Groups 120 is sent to the Inverter 122 which inverts the element in the group to produce k^{-1}. The element k^{-1} together with the received element $k(M)$ is presented to the Group Action 123 which computes $k^{-1}(k(M)) = M$. This is sent to the Canonical Form Module 124 which computes the canonical form in the message space M. Thus the message M has been transferred from A to B.

Note that the above key transport protocol for groups will generally have low bandwidth (provided the bit–length of M is sufficiently large), but the algorithm for canonical forms (Canonical Form Module 123) will very often be much more computationally intensive than the comparison algorithms (Comparison Module 113).

Example 2:

An example of a key transport protocol for monoids is given when the monoid U is the braid group with N generators (see Example 1), $\mathrm{U} = \mathrm{M}$, is the same as the message space, and the action is defined by

$$u(m) = u \cdot m \quad \text{(braid multiplication)} \qquad \text{for all } u \in \mathrm{U}, m \in \mathrm{M}.$$

Note that in this example inverses of elements are not required so that G is viewed as having only the structure of a monoid. A polynomial time algorithm for comparing words in the braid group is given in Patrick Dehornoy, "A fast method for comparing braids," Advances in Mathematics 125 (1997), pages 200–235 and also in Joan S. Birman, Ki Hyoung Ko, and Sang Jin Lee, "A new approach to the word and conjugacy problems in the braid groups," to appear in Advances in Mathematics.

With these choices, the key transport protocol is depicted in figure 11 and Dehorney's or the Birman–Ko–Lee algorithm can be used as a basis for the Compare and Choose Module 114.

Example 3:

An example of a key transport protocol for groups is given when the group G is the braid group with N generators (see Example 1), $G = \mathrm{M}$ is the same as the message space, and the action is defined by braid group conjugation:

$$g(m) = gmg^{-1}, \qquad \text{for all } g \in G, m \in \mathrm{M}.$$

A polynomial time algorithm for computing canonical forms in the braid group is given in Joan S. Birman, Ki Hyoung Ko, and Sang Jin Lee, "A new approach to the word and conjugacy problems in the braid groups," to appear in Advances in Mathematics.

With these choices, the key transport protocol is depicted in figure 12 and the Birman–Ko–Lee algorithm can be used as a basis for the Canonical Form Module 124.

Example 4:

Another example of a key transport protocol for groups is given when G is the braid

group with N generators (see Example 1), M is the free group generated by the set $\{a_1, \ldots, a_N\}$, and the action of G on M is given as follows (see Emil Artin, "Theorie der Zöpfe," Hamb. Abh. 4 (1925), pages 47–72): for $i = 1, \ldots, N$,

$$x_i(a_i) = a_{i+1}, \quad x_i(a_{i+1}) = a_{i+1}^{-1} a_i \, a_{i+1}$$

$$x_i(a_j) = a_j \qquad \text{for } j = 1, \ldots, i-1, i+2, \ldots, N.$$

In this instance the algorithm for the Canonical Form Module 124 is simply free reduction in the free group M, and the algorithm for Group Action 123 is generally exponential in the word length of the acting braid group element.

CLAIMS

1. An encryption system comprising:

a monoid key establishment apparatus responsive to an input monoid, a private element, and a combined input monoid list;

a combinatorial action unit connected to said monoid key establishment apparatus responsive to an input message and having an encrypted output.

2. An encryption system according to claim 1, wherein said monoid key establishment system is group based.

3. An encryption system according to claim 1, wherein said monoid key establishment system is braid group based.

4. An encryption system according to claim 1, wherein said combinatorial action unit is a monoid action unit.

5. An encryption system according to claim 4, further comprising a comparison module referencing said encrypted output to an encrypted input.

6. An encryption system according to claim 2, wherein said combinatorial action unit is a group action unit.

7. An encryption system according to claim 3, wherein said combinatorial action unit is a braid group action unit.

8. An encryption system according to claim 6, wherein said combinatorial action unit further comprises a key inverter.

9. An encryption system according to claim 8, further comprising a canonical form modulator responsive to said encrypted output.

10. An encryption system according to claim 1, further comprising means for creating a set of input monoids.

11. An encryption system according to claim 10, wherein said means for creating a set of input monoids further comprises a monoid processor responsive to a pseudo random number generator.

12. A key agreement system comprising:

a combinatorial group modulator using a private element to act on a group of elements associated with a remote system to generate a local combination;

a combinatorial multiplier responsive to a multiplier input and a remotely generated combination wherein said multiplier input is related to said private element; and

a key extractor responsive to said private element and an output of said combinatorial multiplier.

13. A key agreement system according to claim 12, further comprising:

a combinatorial means for generating a local group of elements; and

a private element generator, wherein said private element is generated from one or more elements of said local group elements.

14. A key agreement system according to claim 12, wherein said combinatorial multiplier is a group multiplier.

15. A key agreement system according to claim 12, wherein said combinatorial group modulator comprises:

a braid group conjugation module; and

a rewriter connected to an output of said braid group conjugation module.

16. A key agreement system according to claim 15, wherein said rewriter is responsive to a pseudorandom number.

17. A key agreement system according to claim 15, wherein said combinatorial multiplier is a braid group multiplier.

18. An encryption method comprising the steps of:

transforming an input monoid, a private element, and a combined input monoid list into a monoid key wherein said transforming is based on the word problem for monoids;

combinatorially acting on said monoid key and an input message to create an encrypted output.

19. An encryption method according to claim 18, wherein said step of acting is based on group theory.

20. An encryption method according to claim 18, wherein said step of combinatorially acting is a monoid action.

21. An encryption method according to claim 20, further comprising the step of comparing said encrypted output to an encrypted input.

22. An encryption method according to claim 19, wherein said step of combinatorially acting is a group action based on the conjugacy problem.

23. An encryption method according to claim 17, wherein said step of combinatorially acting is a braid group action.

24. An encryption method according to claim 22, wherein said step of combinatorially acting further comprises a key inversion step.

25. An encryption method according to claim 24, further comprising the step of canonically reformatting said encrypted output.

26. An encryption method according to claim 18, further comprising the step of creating a set of input monoids.

27. An encryption method according to claim 26, wherein said step of creating a set of input monoids further comprises the step of processing pseudo random numbers into monoids.

28. A method for establishing a key comprising the steps of:

transforming a private element and a group of elements associated with a remote system into a local combination based on a combinatorial relationship;

combinatorially multiplying a multiplier input and a remotely generated combination wherein said multiplier input is related to said private element; and

extracting a key from said private element and the result of the step of combinatorially multiplying.

29. A method according to claim 28, wherein said step of combinatorially multiplying further comprises the step of rewriting the result responsive to a pseudorandom input.

30. A method for establishing a key according to claim 28, further comprising:

a step of generating a local group of elements; and

a step of generating the private element, wherein said private element is generated from one or more elements of said local group elements.

31. A method for establishing a key according to claim 28, wherein said step of combinatorially multiplying is a group multiplication.

32. A method for establishing a key according to claim 28, wherein said step of transforming comprises the steps of:

conjugating a combination based on a combinatorial relation by said private element; and

rewriting, responsive to a pseudorandom process, the result of the step of conjugating.

33. A method for establishing a key according to claim 32, wherein said step of combinatorially multiplying is a braid group multiplication.

34. A method for establishing a key according to claim 32, wherein the step of conjugating is a braid group conjugation.

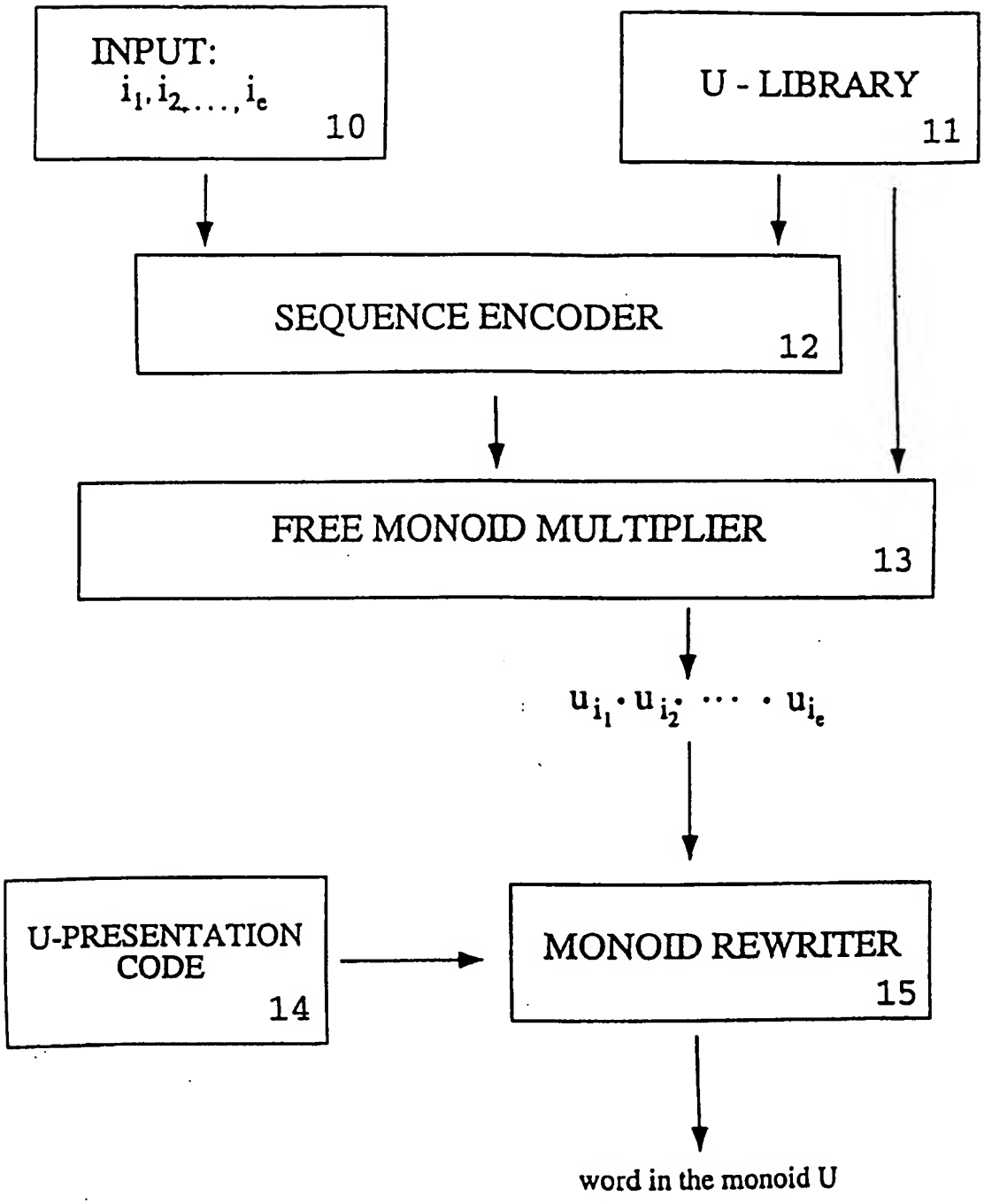

INPUT:
$i_1, i_2, \ldots, i_e$
10
U - LIBRARY
11
SEQUENCE ENCODER
12
FREE MONOID MULTIPLIER
13
$u_{i_1} \cdot u_{i_2} \cdot \cdots \cdot u_{i_e}$
U-PRESENTATION CODE
14
MONOID REWRITER
15
word in the monoid U

figure 1

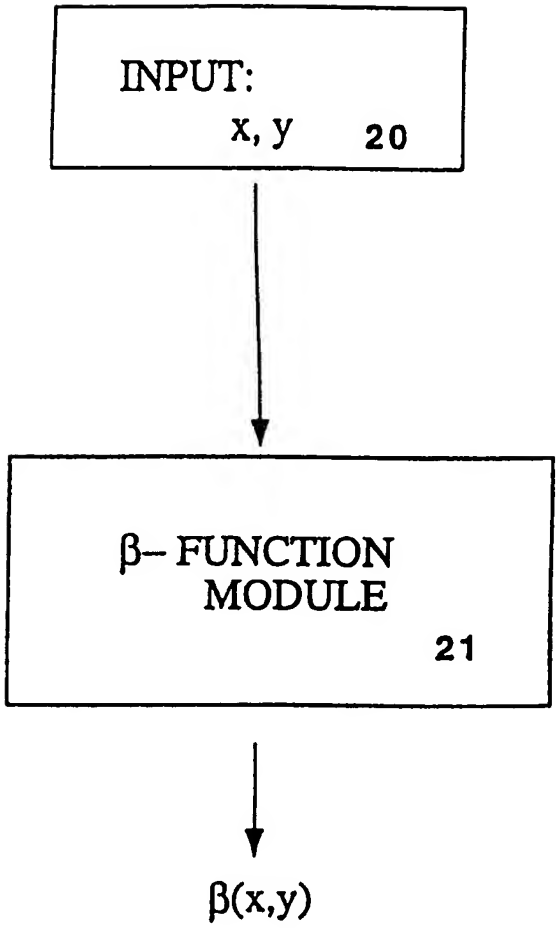


figure 2

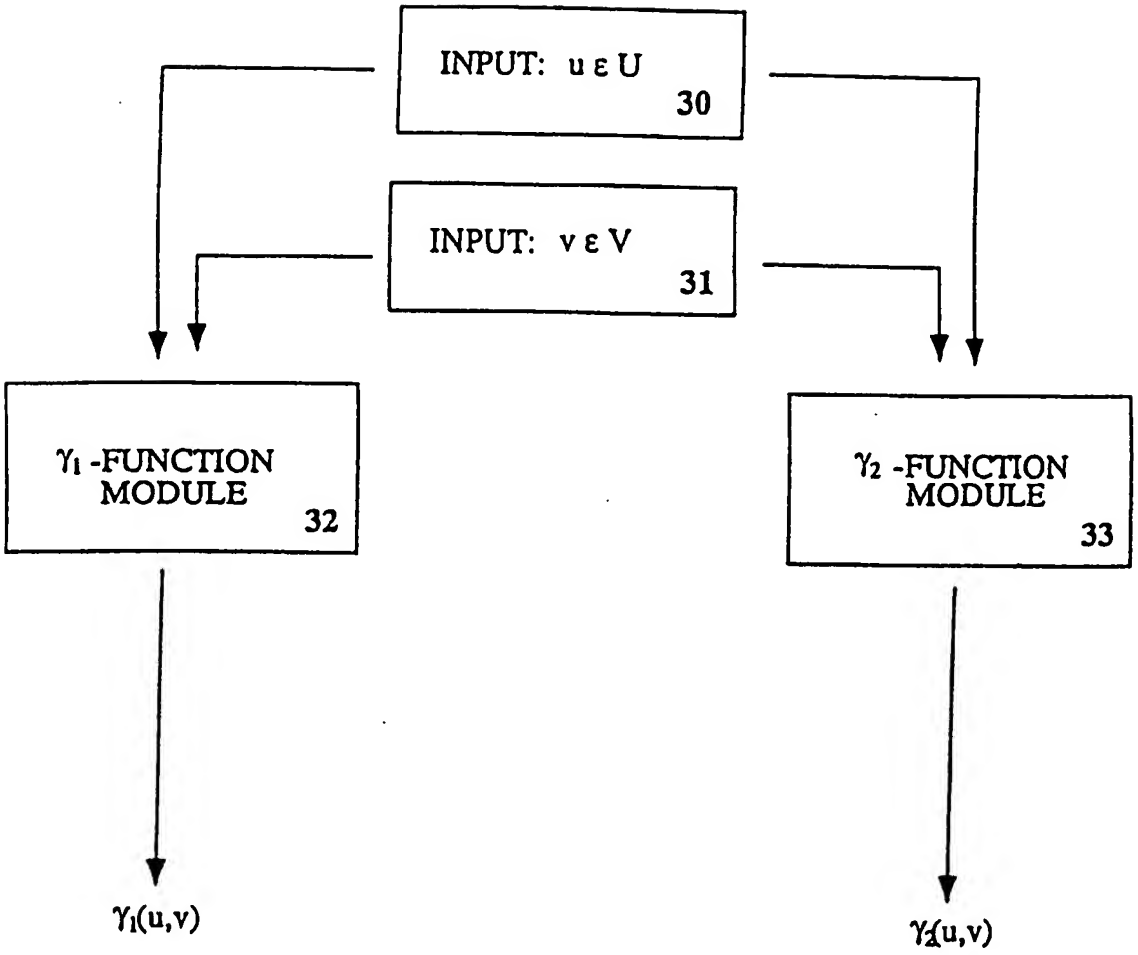

INPUT: u ε U
30
INPUT: v ε V
31
γ1 -FUNCTION MODULE
32
γ2 -FUNCTION MODULE
33
γ1(u,v)
γ2(u,v)

figure 3

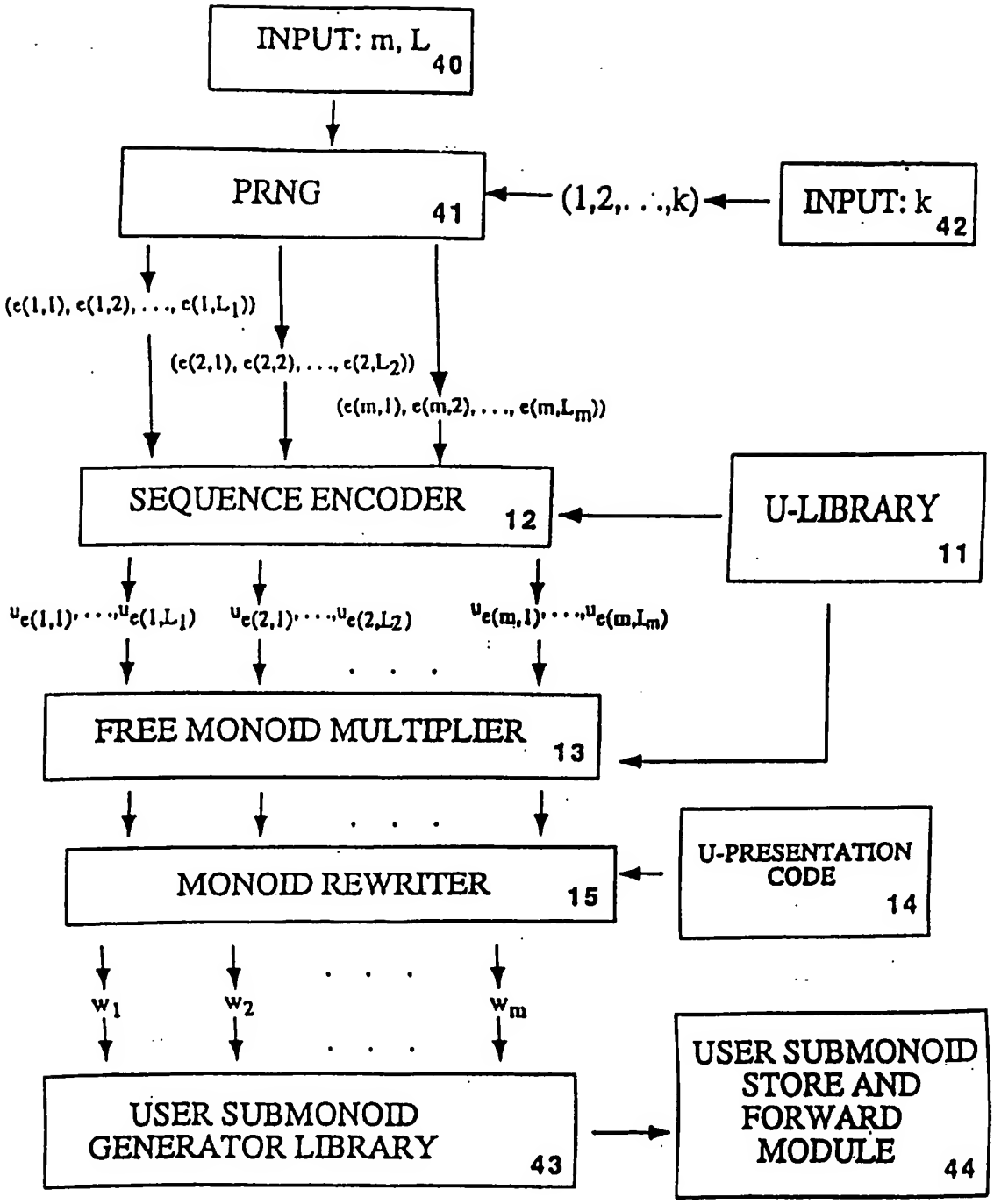

INPUT: m, L 40
PRNG 41
(1,2,. . .,k)
INPUT: k 42
(e(1,1), e(1,2), . . ., e(1,L1))
(e(2,1), e(2,2), . . ., e(2,L2))
(e(m,1), e(m,2), . . ., e(m,Lm))
SEQUENCE ENCODER 12
U-LIBRARY 11
ue(1,1), . . ., ue(1,L1)
ue(2,1), . . ., ue(2,L2)
ue(m,1), . . ., ue(m,Lm)
FREE MONOID MULTIPLIER 13
MONOID REWRITER 15
U-PRESENTATION CODE 14
w1
w2
wm
USER SUBMONOID GENERATOR LIBRARY 43
USER SUBMONOID STORE AND FORWARD MODULE 44

figure 4

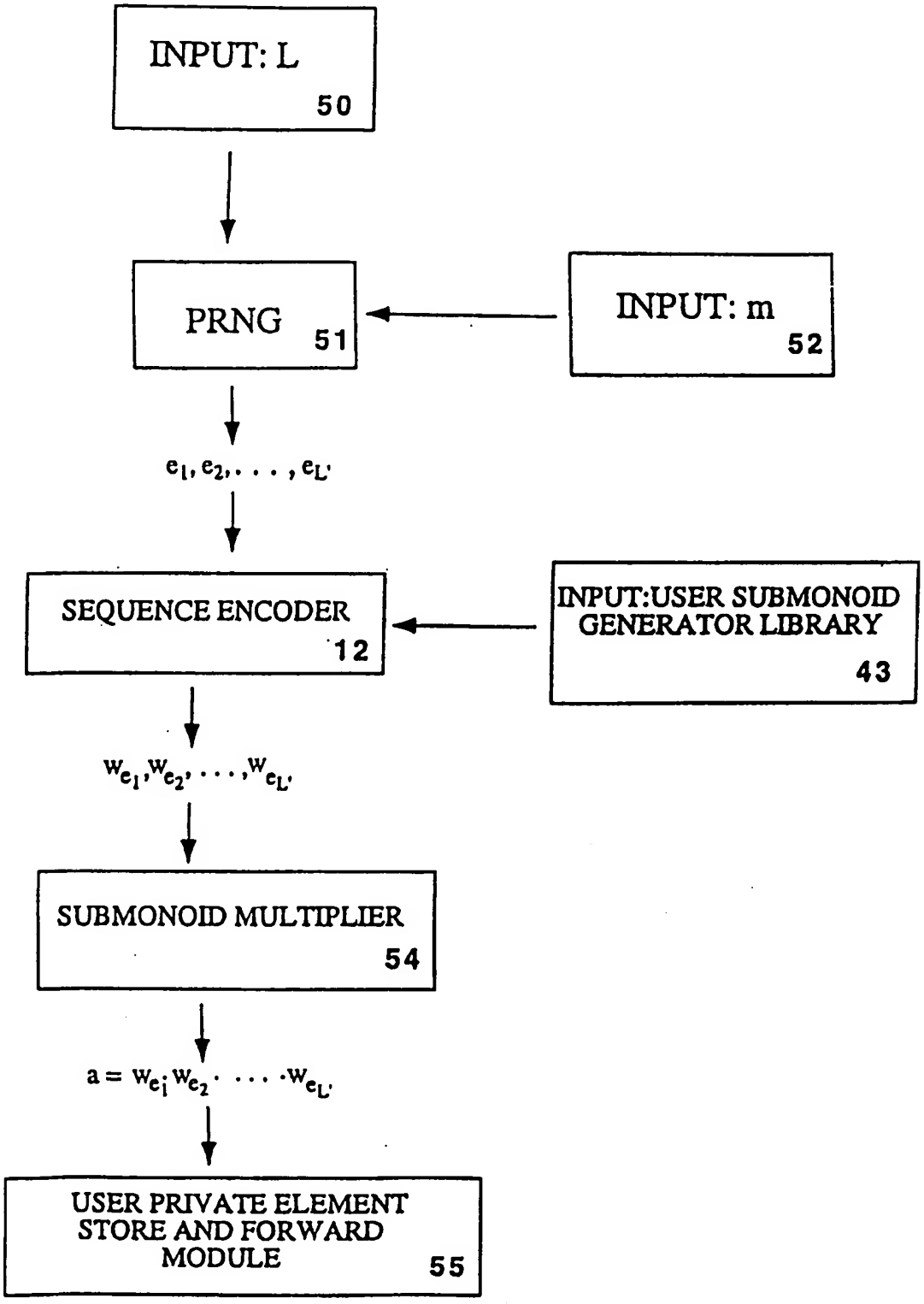

INPUT: L
50
PRNG
51
INPUT: m
52
$e_1, e_2, \ldots, e_{L'}$
SEQUENCE ENCODER
12
INPUT:USER SUBMONOID GENERATOR LIBRARY
43
$w_{e_1}, w_{e_2}, \ldots, w_{e_{L'}}$
SUBMONOID MULTIPLIER
54
$a = w_{e_1} w_{e_2} \cdot \ldots \cdot w_{e_{L'}}$
USER PRIVATE ELEMENT STORE AND FORWARD MODULE
55

figure 5

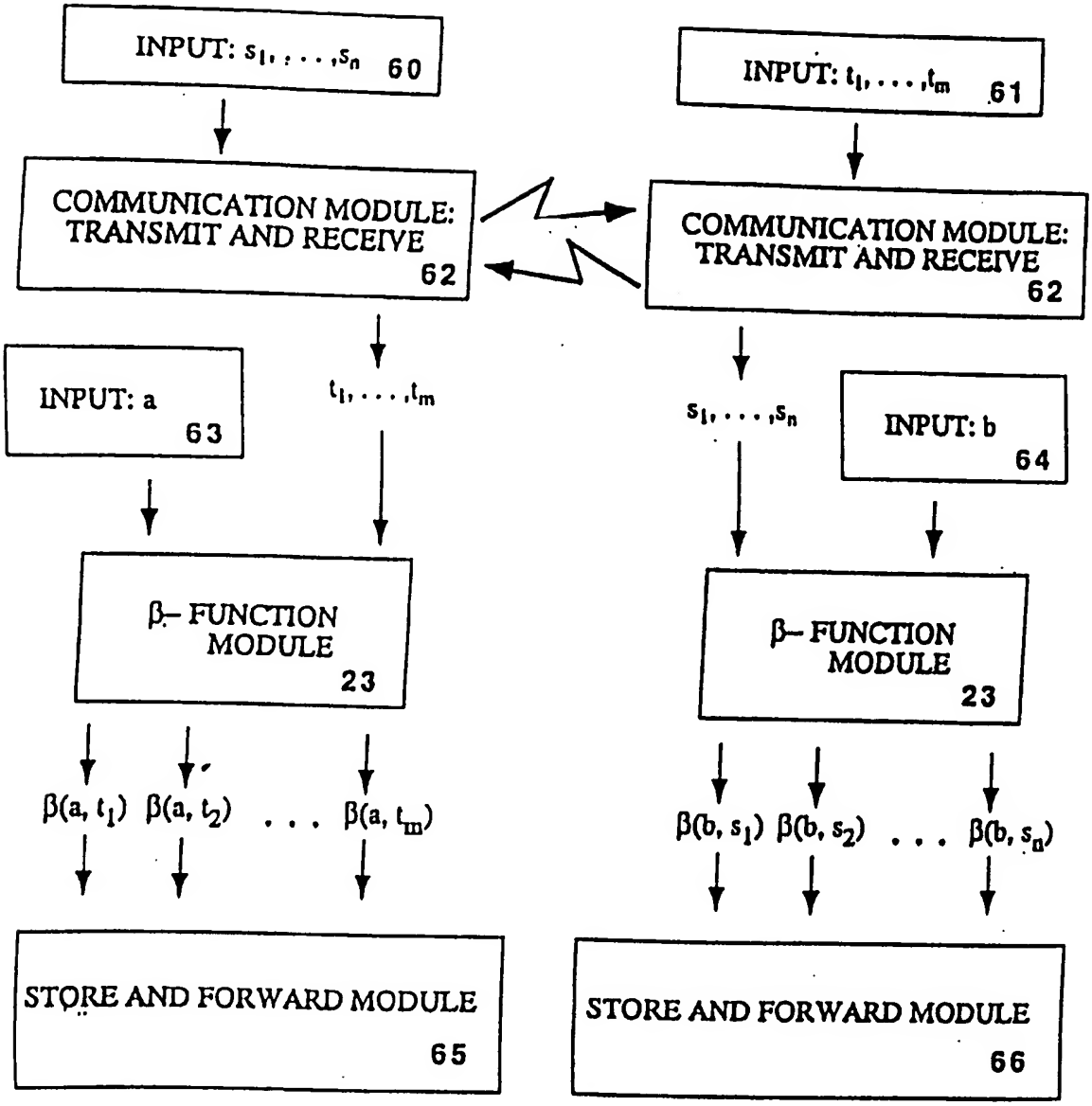

INPUT: $s_1, \ldots, s_n$ 60
INPUT: $t_1, \ldots, t_m$ 61
COMMUNICATION MODULE: TRANSMIT AND RECEIVE 62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE 62
INPUT: a 63
$t_1, \ldots, t_m$
$s_1, \ldots, s_n$
INPUT: b 64
β– FUNCTION MODULE 23
β– FUNCTION MODULE 23
$\beta(a, t_1)$ $\beta(a, t_2)$ . . . $\beta(a, t_m)$
$\beta(b, s_1)$ $\beta(b, s_2)$ . . . $\beta(b, s_n)$
STORE AND FORWARD MODULE 65
STORE AND FORWARD MODULE 66

figure 6

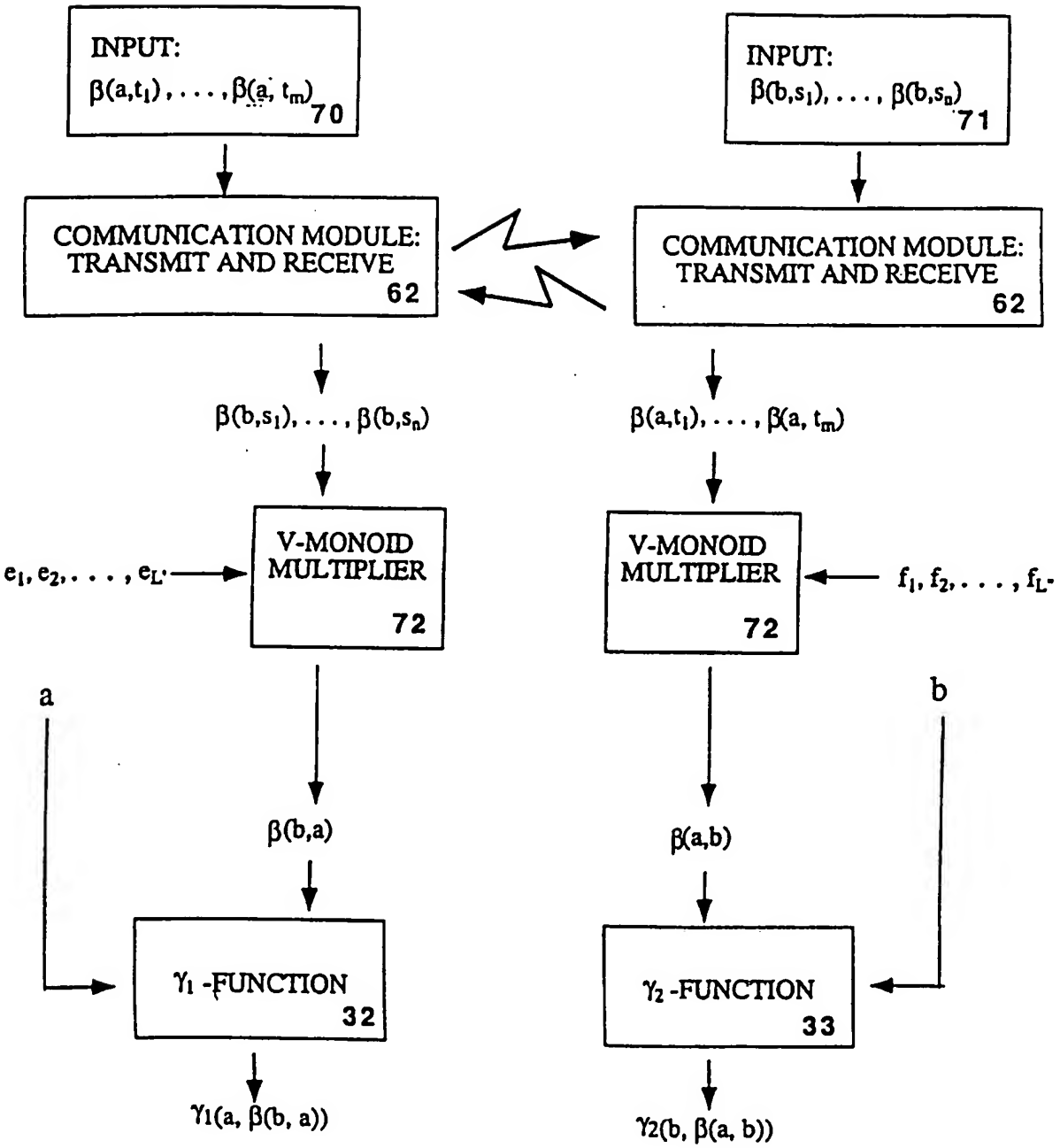

INPUT:
β(a,t1), . . . , β(a, tm)
70
INPUT:
β(b,s1), . . . , β(b,sn)
71
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
β(b,s1), . . . , β(b,sn)
β(a,t1), . . . , β(a, tm)
V-MONOID MULTIPLIER
72
V-MONOID MULTIPLIER
72
e1, e2, . . . , eL'
f1, f2, . . . , fL'
a
b
β(b,a)
β(a,b)
γ1 -FUNCTION
32
γ2 -FUNCTION
33
γ1(a, β(b, a))
γ2(b, β(a, b))

figure 7

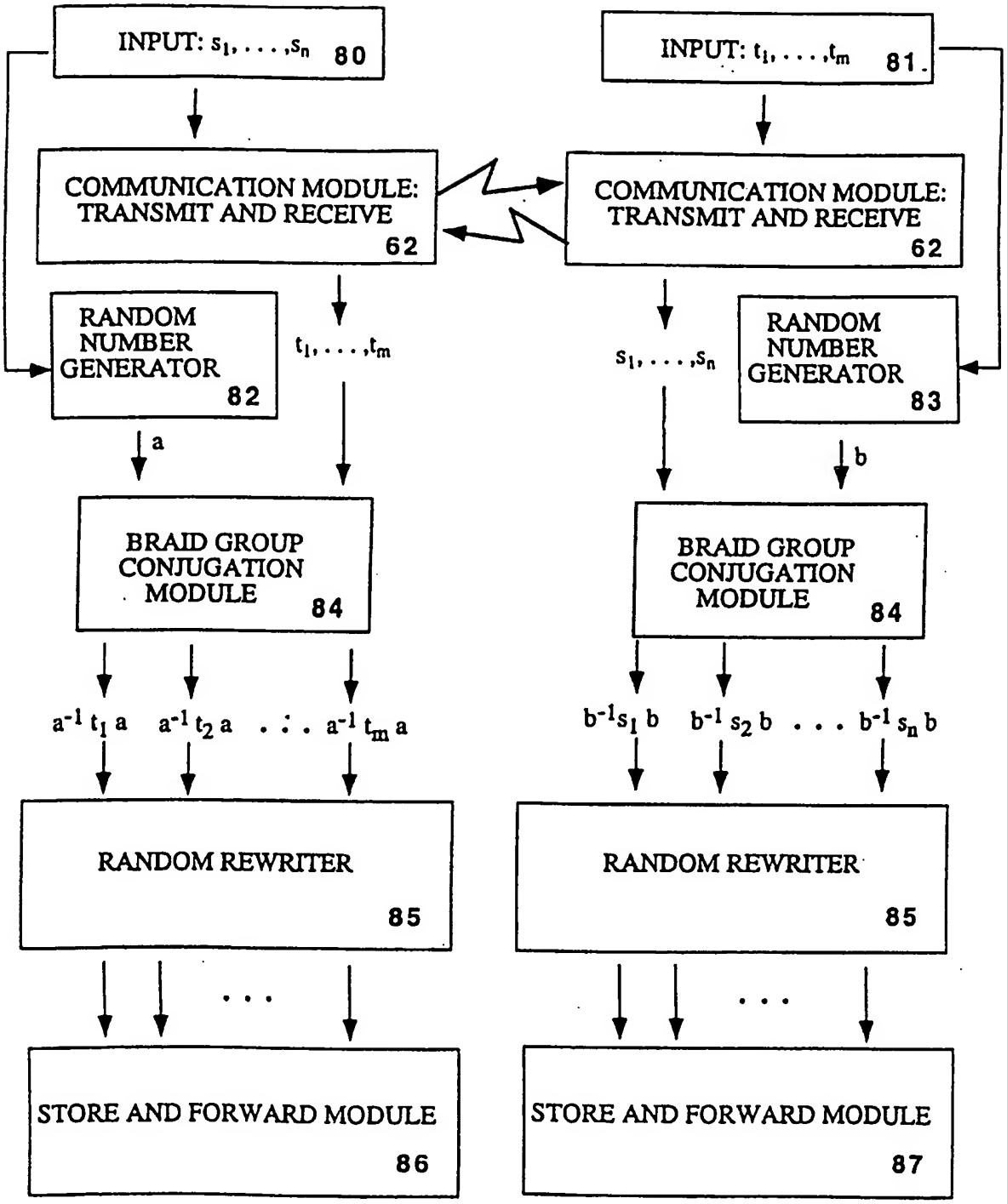

INPUT: $s_1, \ldots, s_n$ 80
INPUT: $t_1, \ldots, t_m$ 81
COMMUNICATION MODULE: TRANSMIT AND RECEIVE 62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE 62
RANDOM NUMBER GENERATOR 82
$t_1, \ldots, t_m$
$s_1, \ldots, s_n$
RANDOM NUMBER GENERATOR 83
a
b
BRAID GROUP CONJUGATION MODULE 84
BRAID GROUP CONJUGATION MODULE 84
$a^{-1} t_1 a$ $a^{-1} t_2 a$ . . . $a^{-1} t_m a$
$b^{-1} s_1 b$ $b^{-1} s_2 b$ . . . $b^{-1} s_n b$
RANDOM REWRITER 85
RANDOM REWRITER 85
. . .
. . .
STORE AND FORWARD MODULE 86
STORE AND FORWARD MODULE 87

figure 8

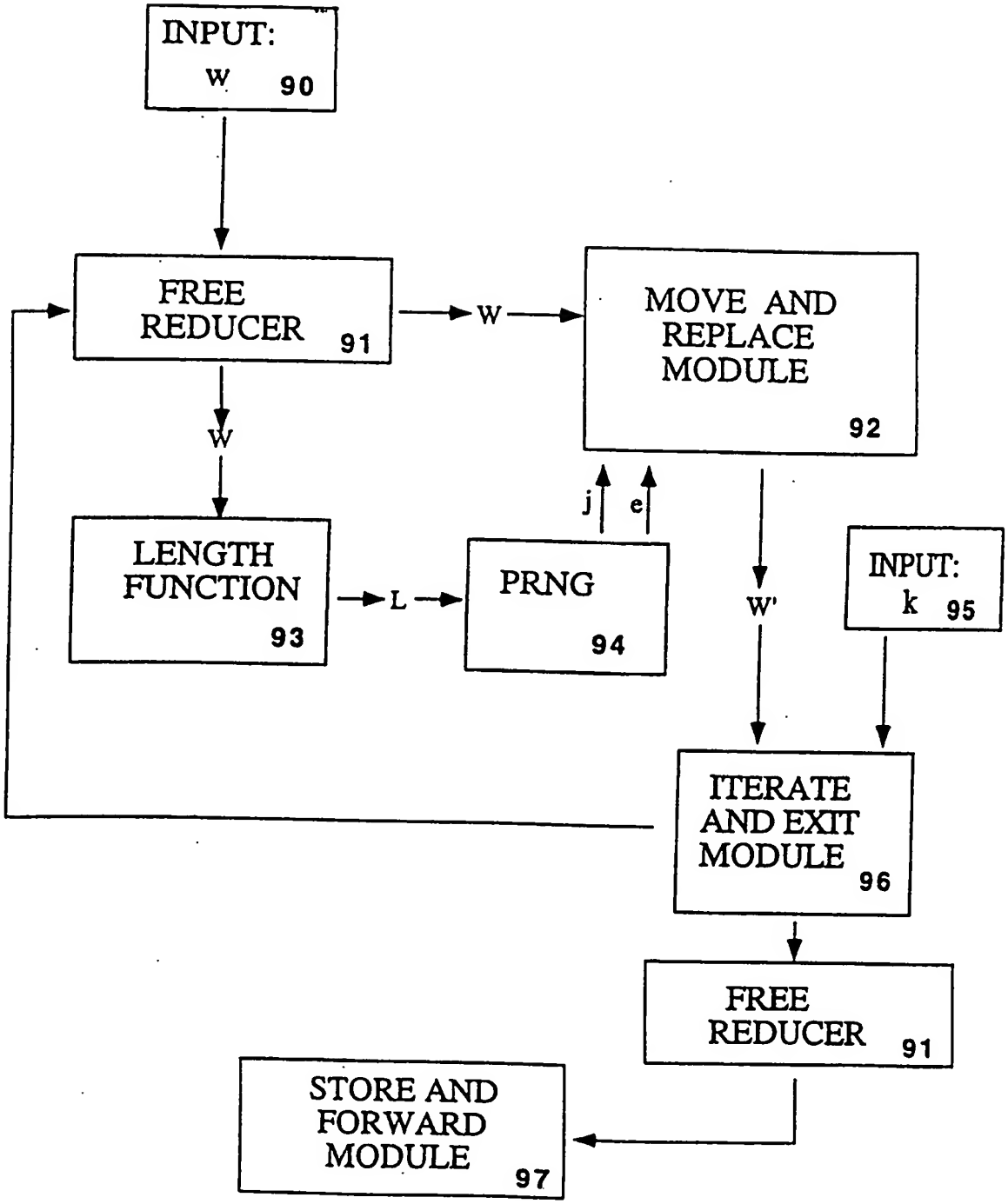

INPUT:
w 90
FREE REDUCER 91
w
MOVE AND REPLACE MODULE 92
w
j
e
LENGTH FUNCTION 93
L
PRNG 94
w'
INPUT:
k 95
ITERATE AND EXIT MODULE 96
FREE REDUCER 91
STORE AND FORWARD MODULE 97

figure 9

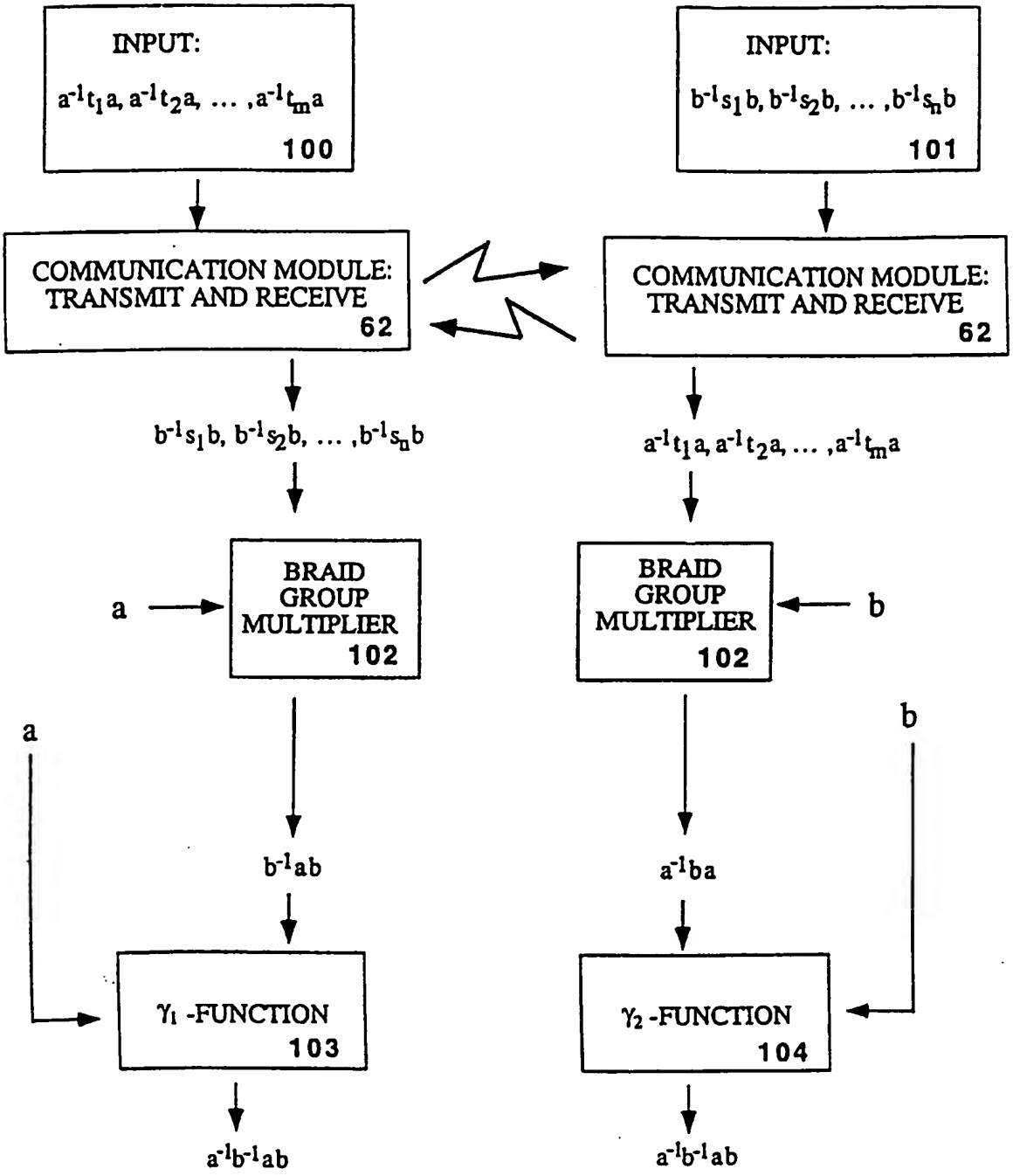

INPUT:
$a^{-1}t_1a, a^{-1}t_2a, \ldots, a^{-1}t_ma$
100
INPUT:
$b^{-1}s_1b, b^{-1}s_2b, \ldots, b^{-1}s_nb$
101
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
$b^{-1}s_1b, b^{-1}s_2b, \ldots, b^{-1}s_nb$
$a^{-1}t_1a, a^{-1}t_2a, \ldots, a^{-1}t_ma$
a
BRAID GROUP MULTIPLIER
102
BRAID GROUP MULTIPLIER
102
b
a
b
$b^{-1}ab$
$a^{-1}ba$
γ_1 -FUNCTION
103
γ_2 -FUNCTION
104
$a^{-1}b^{-1}ab$
$a^{-1}b^{-1}ab$

figure 10

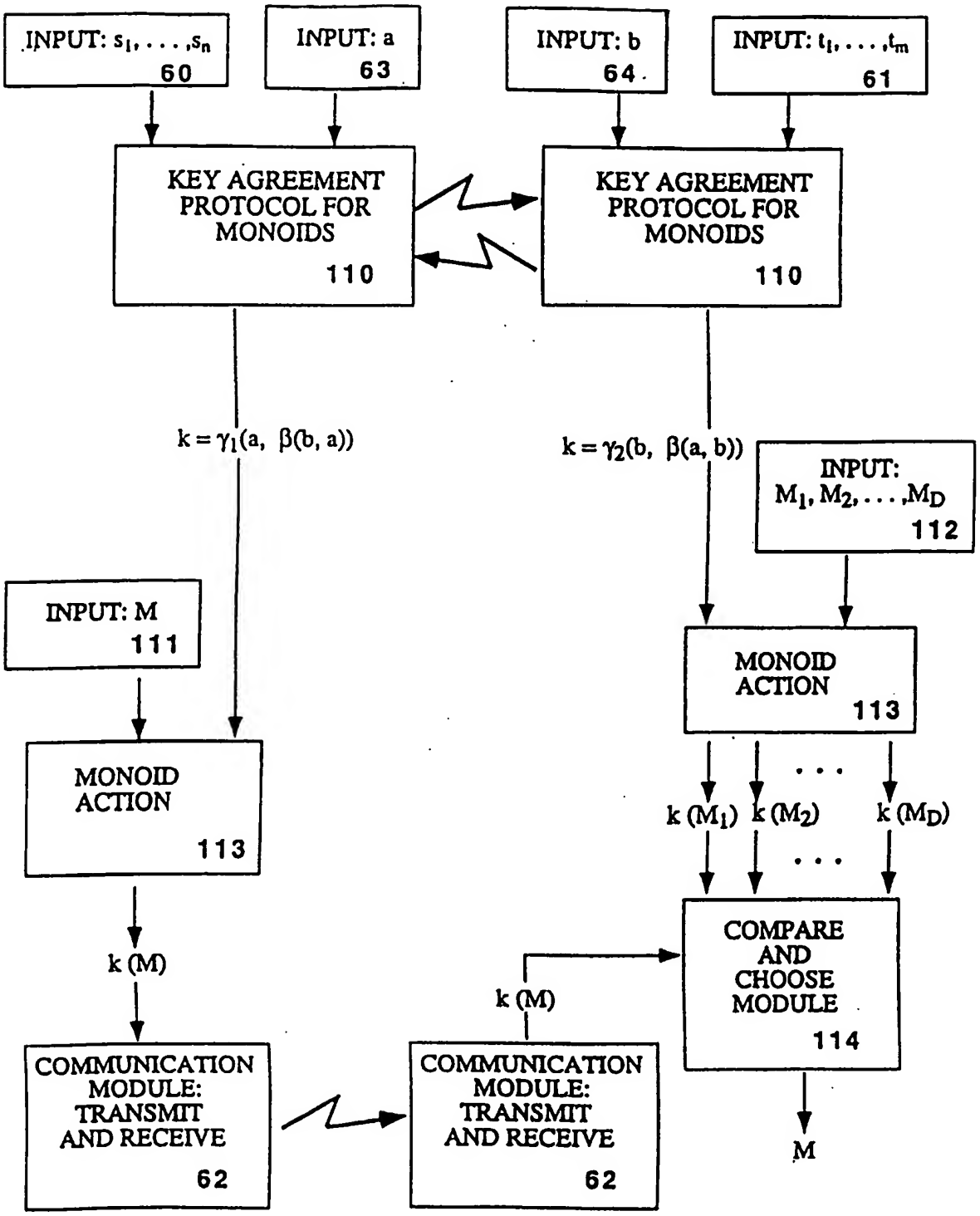

INPUT: $s_1, \ldots, s_n$
60
INPUT: a
63
INPUT: b
64
INPUT: $t_1, \ldots, t_m$
61
KEY AGREEMENT PROTOCOL FOR MONOIDS
110
KEY AGREEMENT PROTOCOL FOR MONOIDS
110
$k = \gamma_1(a, \beta(b, a))$
$k = \gamma_2(b, \beta(a, b))$
INPUT: $M_1, M_2, \ldots, M_D$
112
INPUT: M
111
MONOID ACTION
113
MONOID ACTION
113
k (M)
k (M)
$k(M_1)$ $k(M_2)$ $k(M_D)$
COMPARE AND CHOOSE MODULE
114
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
M

figure 11

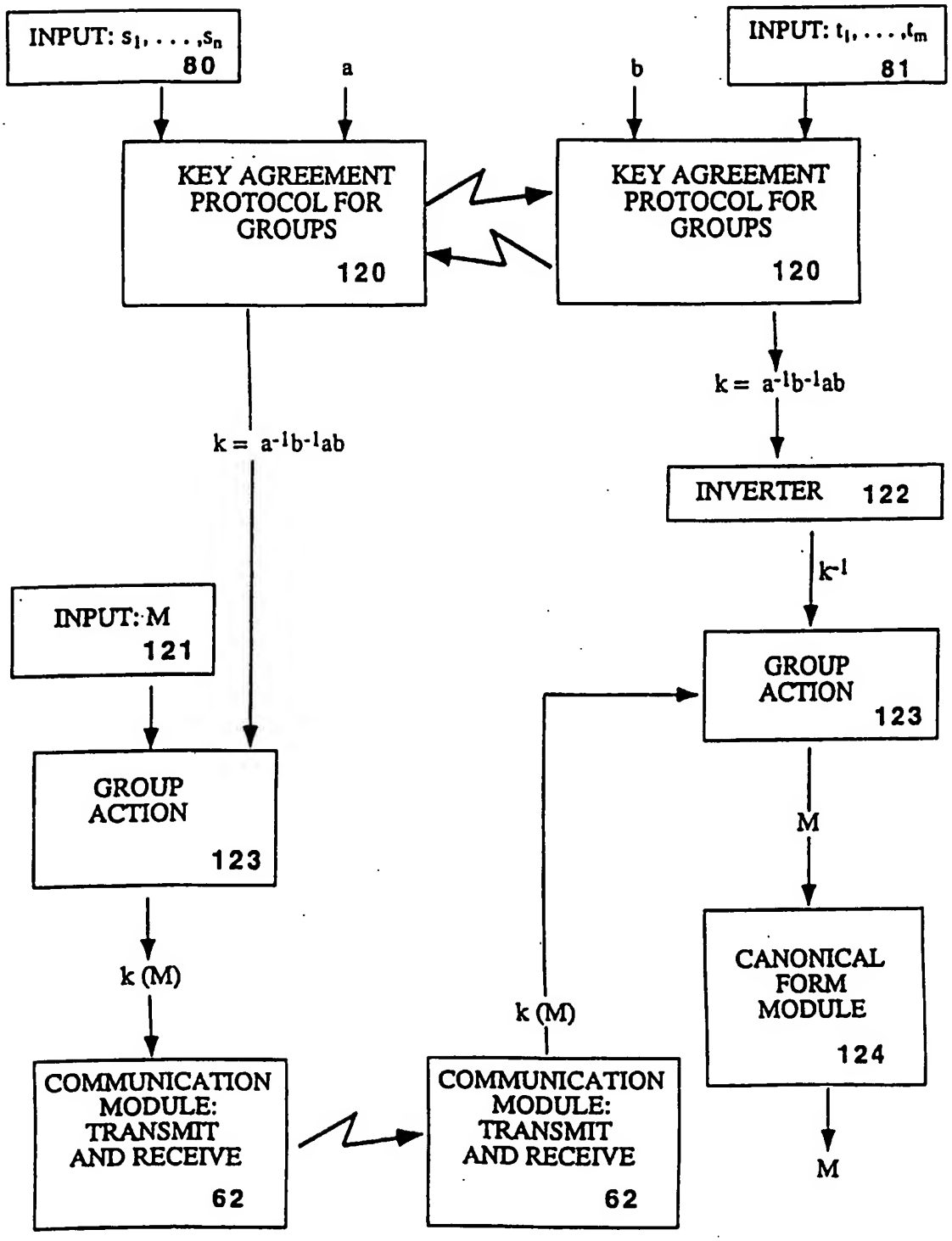

INPUT: $s_1, \ldots, s_n$
80
a
b
INPUT: $t_1, \ldots, t_m$
81
KEY AGREEMENT PROTOCOL FOR GROUPS
120
KEY AGREEMENT PROTOCOL FOR GROUPS
120
$k = a^{-1}b^{-1}ab$
$k = a^{-1}b^{-1}ab$
INVERTER 122
k^{-1}
INPUT: M
121
GROUP ACTION
123
GROUP ACTION
123
M
k (M)
k (M)
CANONICAL FORM MODULE
124
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
COMMUNICATION MODULE: TRANSMIT AND RECEIVE
62
M

figure 12

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/28 — A3

(11) International Publication Number: WO 99/44324

(43) International Publication Date: 2 September 1999 (02.09.99)

(21) International Application Number: PCT/US99/04126

(22) International Filing Date: 25 February 1999 (25.02.99)

(30) Priority Data:
09/030,935 26 February 1998 (26.02.98) US

(71) Applicant: ARITHMETICA, INC. [US/US]; Corporation Trust Center, 1209 Orange Street, Wilimington, DE 19801 (US).

(72) Inventors: ANSHEL, Iris; 31 Peter Lynas Court, Tenafly, NJ 07670 (US). ANSHEL, Michael, M.; Apartment 3C, 1140 Fifth Avenue, New York, NY 10128 (US). GOLDFELD, Dorian; 31 Peter Lynas Court, Tenafly, NJ 07670 (US).

(74) Agents: KOCH, Robert, J. et al.; Fulbright & Jaworski L.L.P., 801 Pennsylvania Avenue, N.W., Washington, DC 20004 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

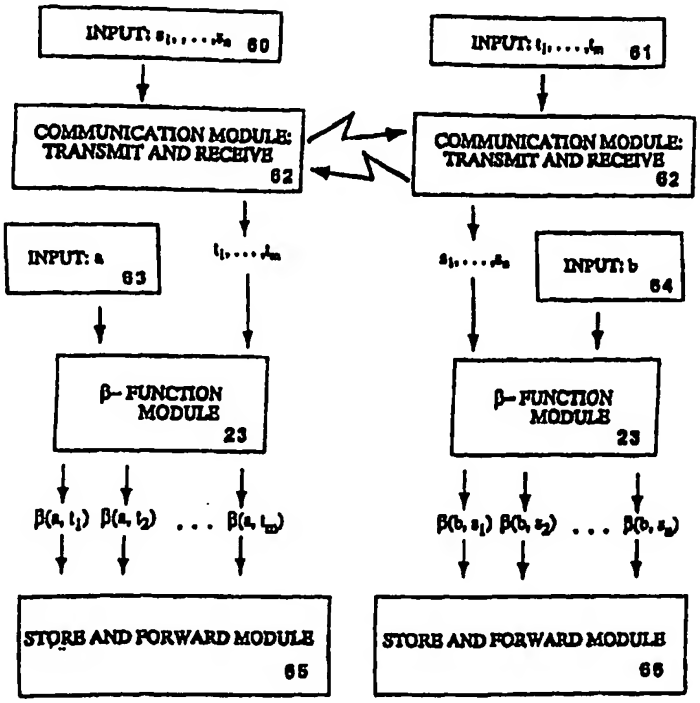
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(88) Date of publication of the international search report:
7 October 1999 (07.10.99)

(54) Title: A METHOD AND APPARATUS FOR CRYPTOGRAPHICALLY SECURE ALGEBRAIC KEY ESTABLISHMENT PROTOCOLS

(57) Abstract

The present invention is a method and apparatus for providing cryptographically secure algebraic key establishment protocols that use monoids and groups possessing certain algorithmic properties. Special fast algorithms associated with certain monoids and groups are used to optimize both key (63) agreement and key transport protocols (62). The cryptographic security of the algorithms (23) is based on the difficulty of solving the conjugacy problem in groups and other known hard algebraic problems. Braid groups and their associated algorithms are the basis for highly rapid key agreement and key transport protocols which employ modest computational resources.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/04126

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/28
US CL :380/28

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/28, 30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS
search terms: braid group, key, monoid, random number generator, combinatorial, encrypt, cipher, cypher, encipher, encypher

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,627,893 A (DEMYTKO) 06 May 1997, col.1, lines 26-40, col.2, lines 4-25, col. 3, lines 29-60.	1-34
Y	US 4,405,829 A (RIVEST et al.) 20 September 1983, col.2, lines 10-62, col.4, lines 4-61, col.5,lines 1-17.	1-34

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"B" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 JULY 1999	17 AUG 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-0040	Authorized officer GAIL O. HAYES Telephone No. (703) 305-9711

Form PCT/ISA/210 (second sheet)(July 1992)*